

FIG. 1

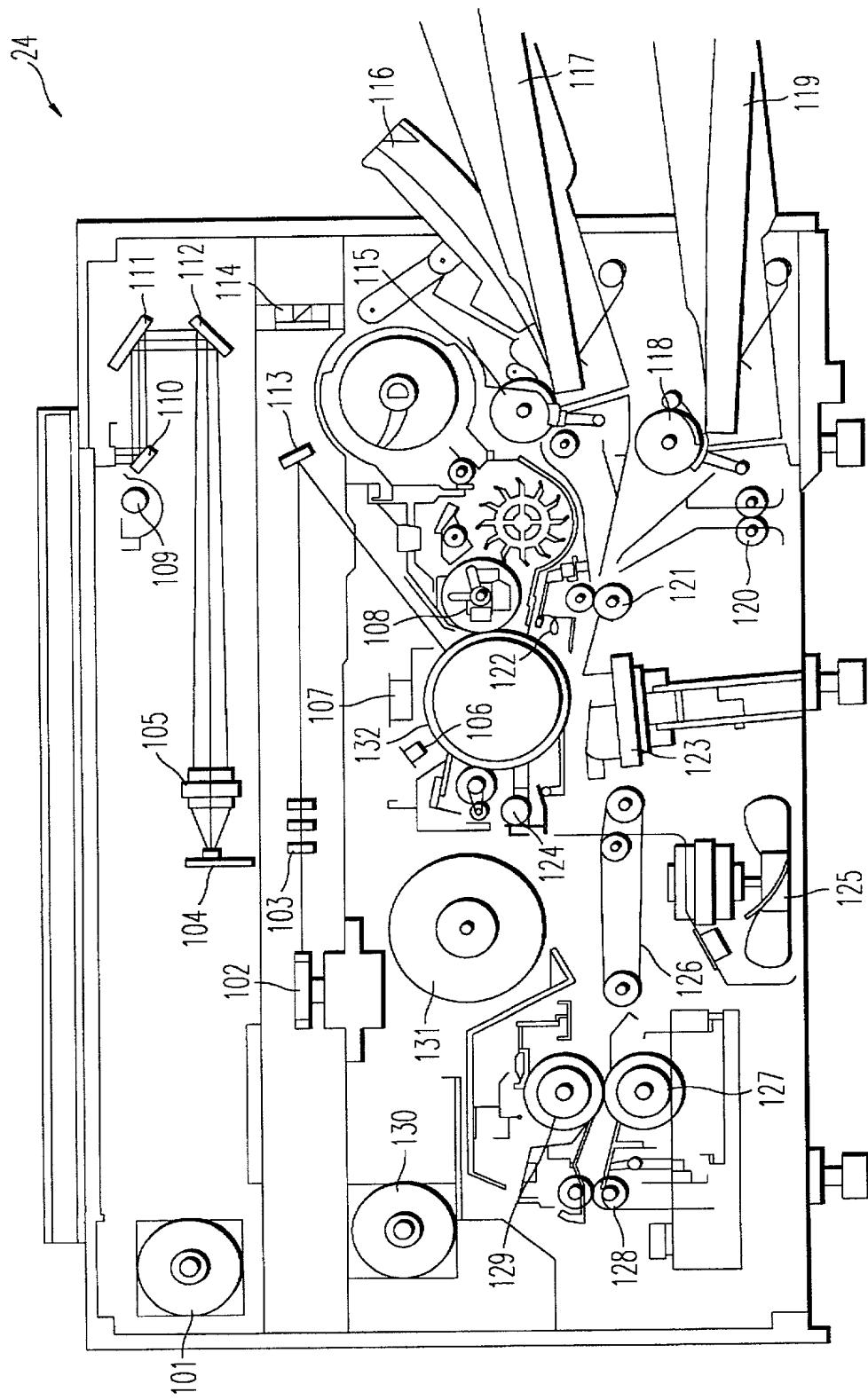


FIG. 2

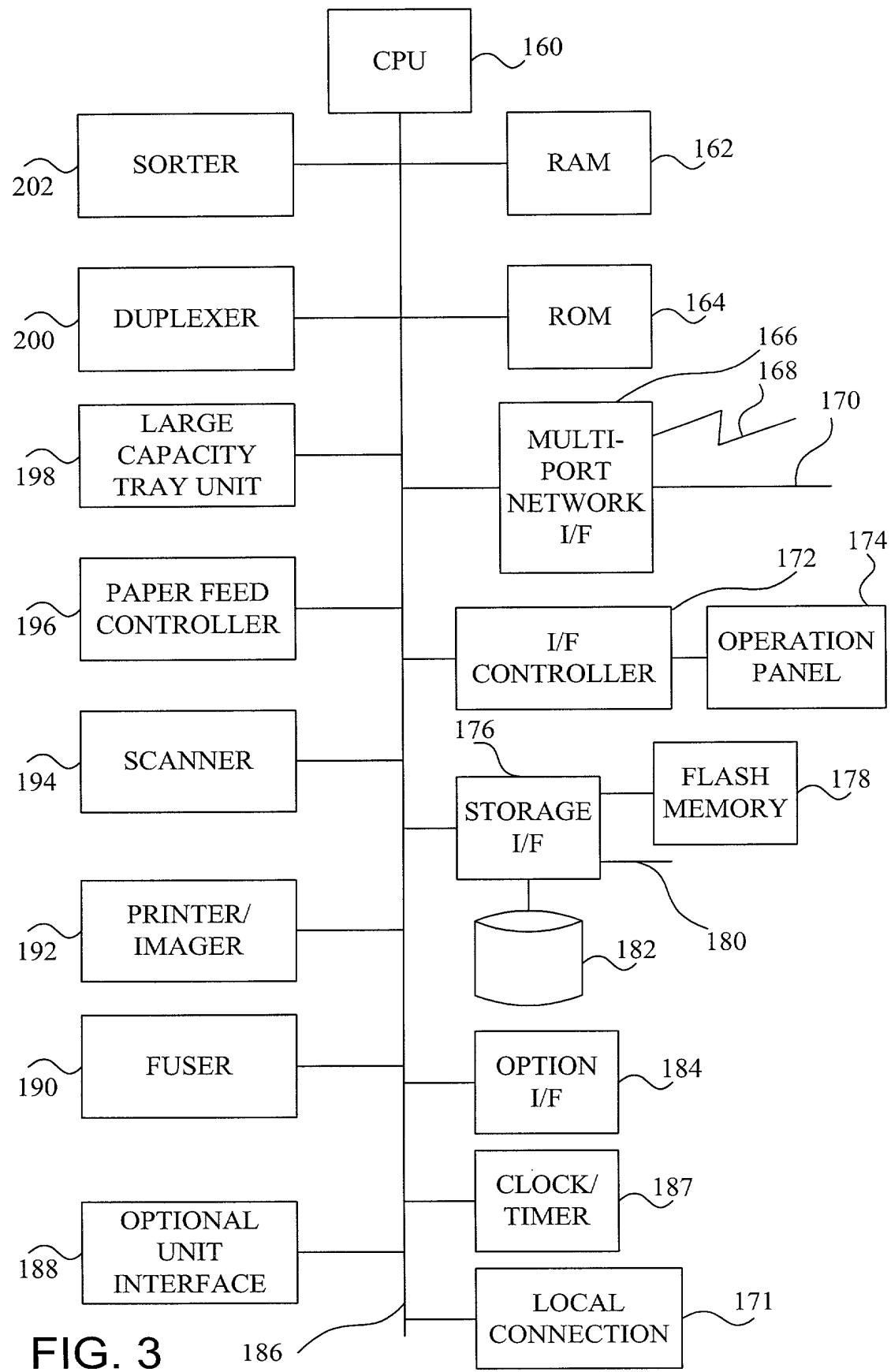


FIG. 3

186

171

187

LOCAL  
CONNECTION

CLOCK/  
TIMER

OPTION  
I/F

STORAGE  
I/F

FLASH  
MEMORY

I/F  
CONTROLLER

MULTI-  
PORT  
NETWORK  
I/F

ROM

RAM

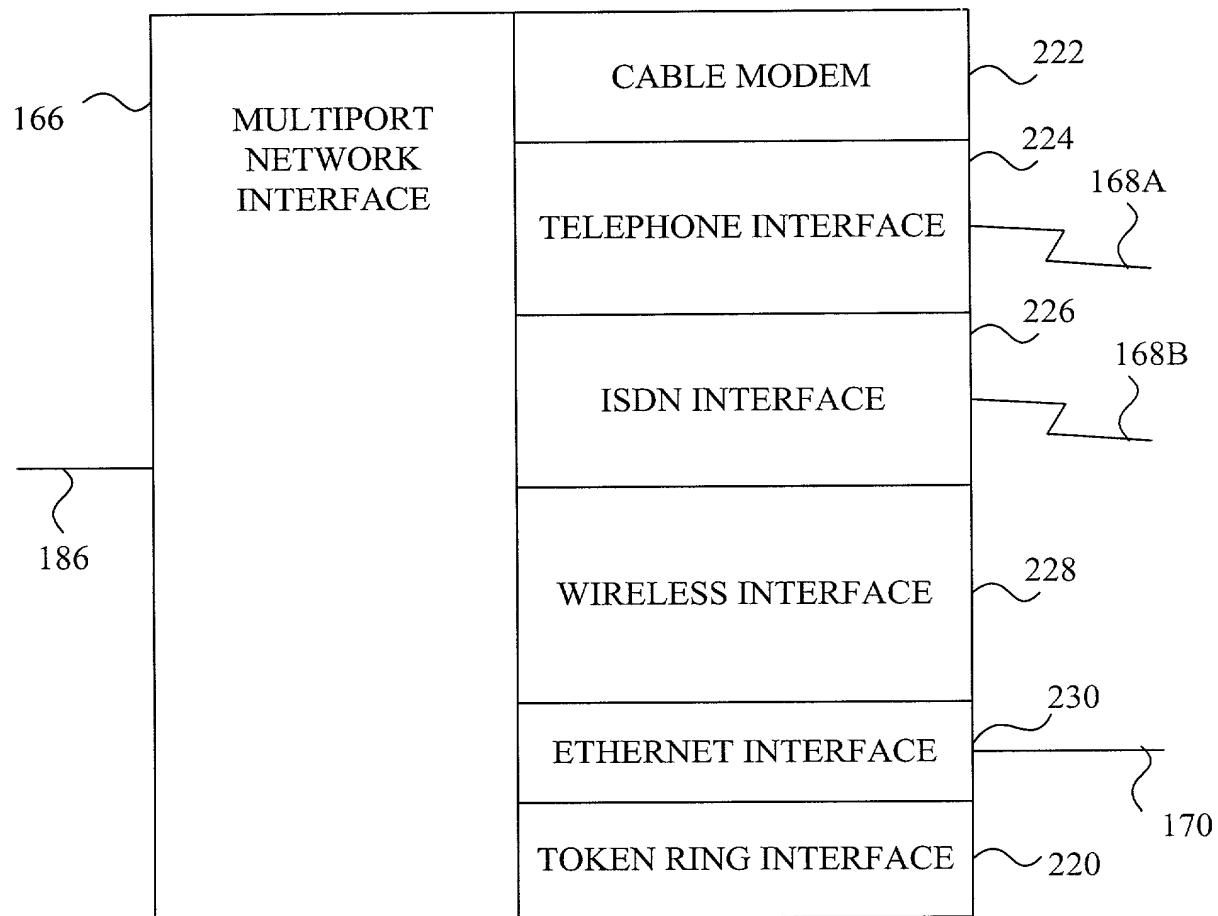
CPU

OBLON, SPIVAK, ET AL

DOCKET #:194539US-2

INV: TETSURO MOTOYAMA ET AL.

SHEET 3 OF 32



**FIG. 4**

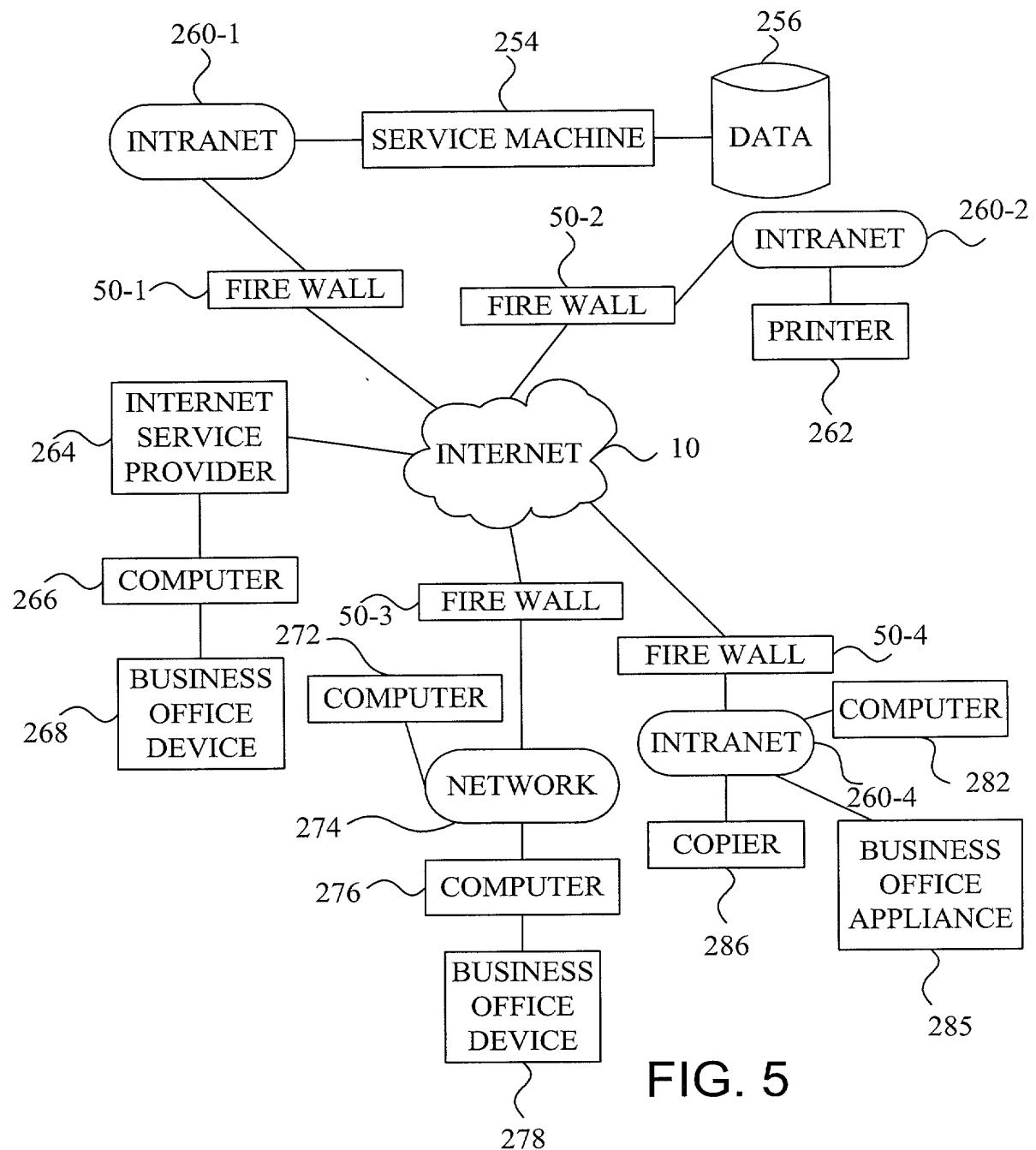


FIG. 5

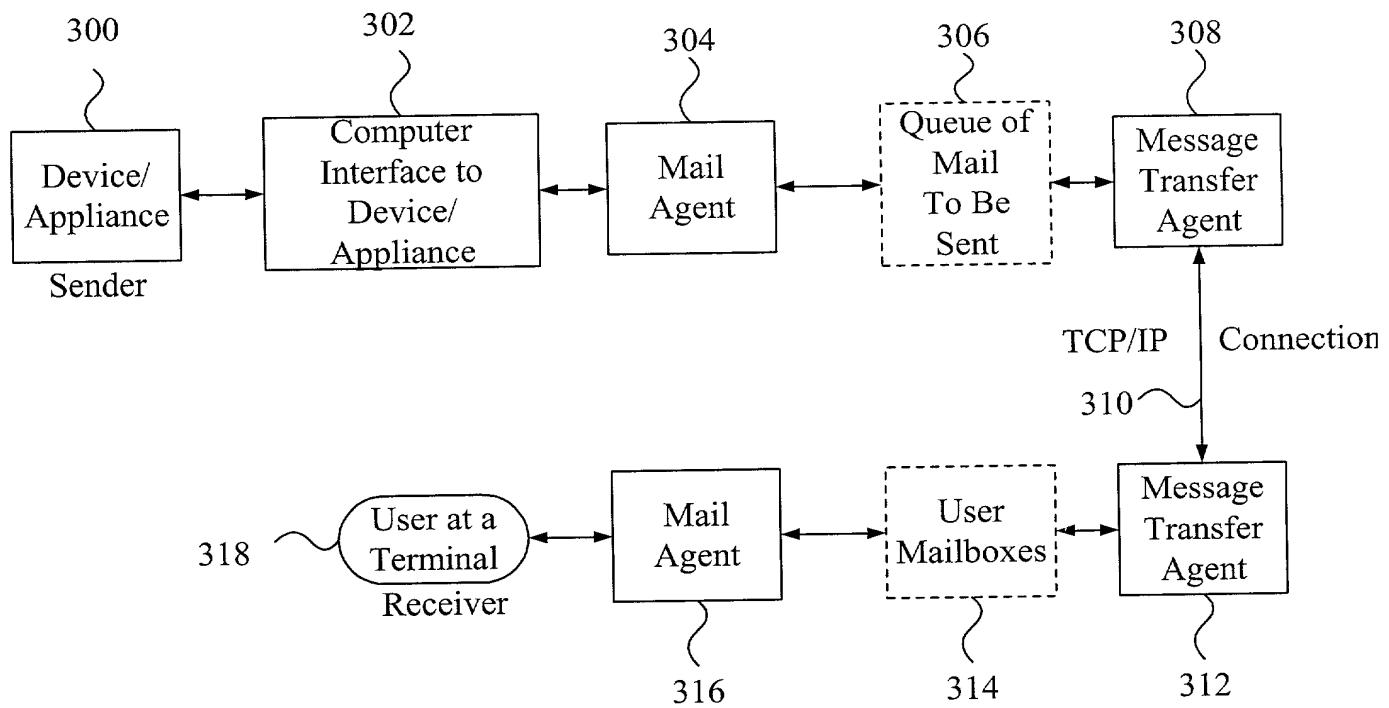


FIG. 6A

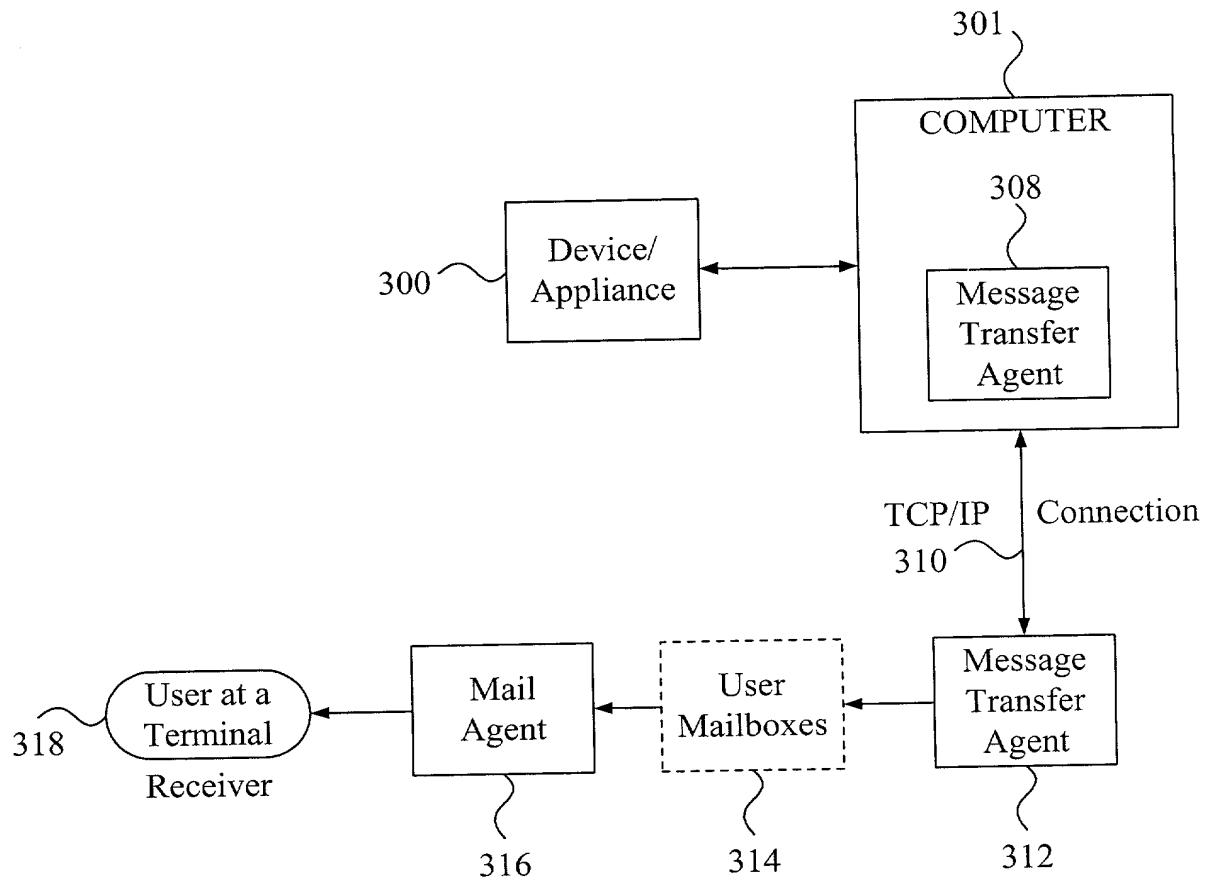


FIG. 6B

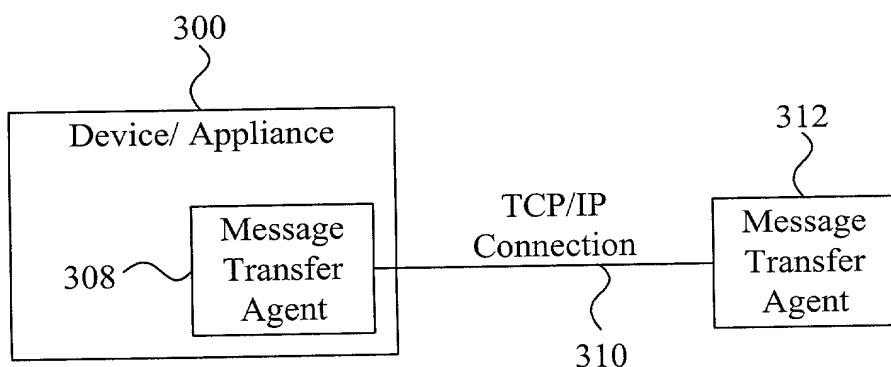


FIG. 6C

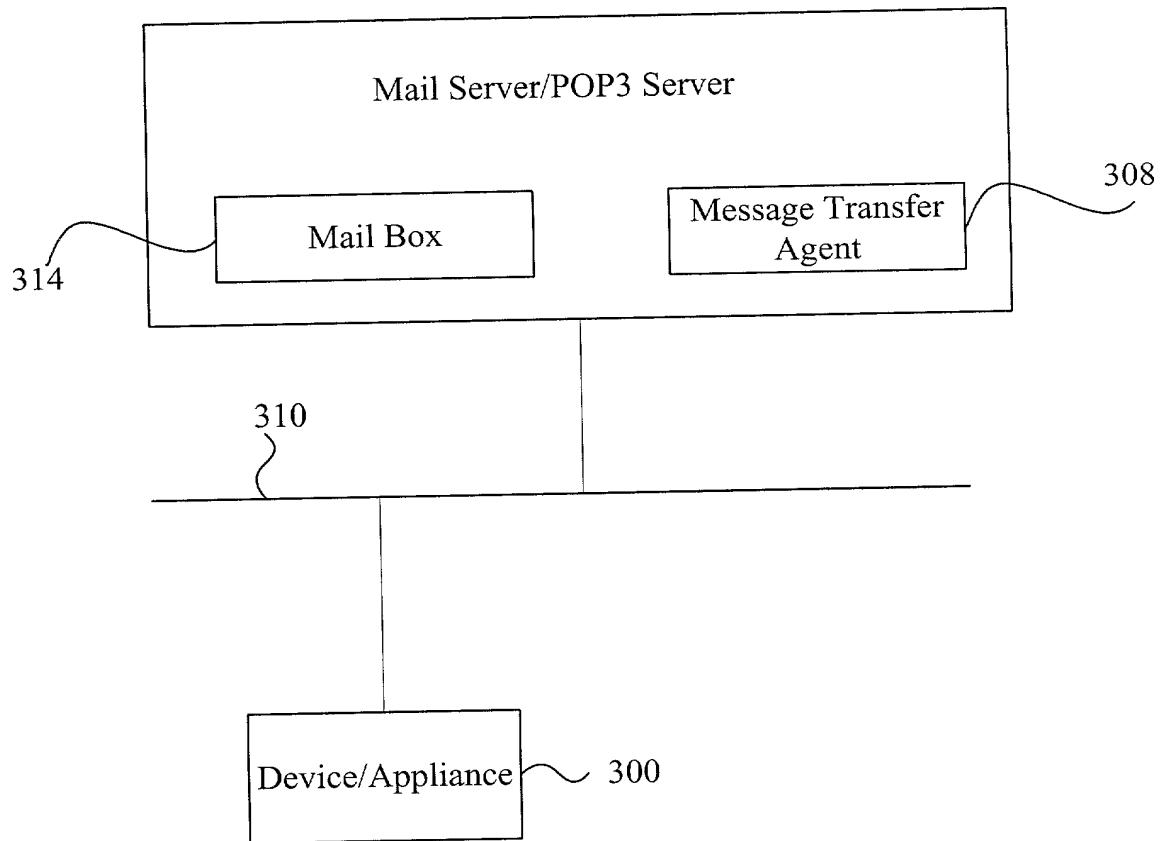


Fig. 6D

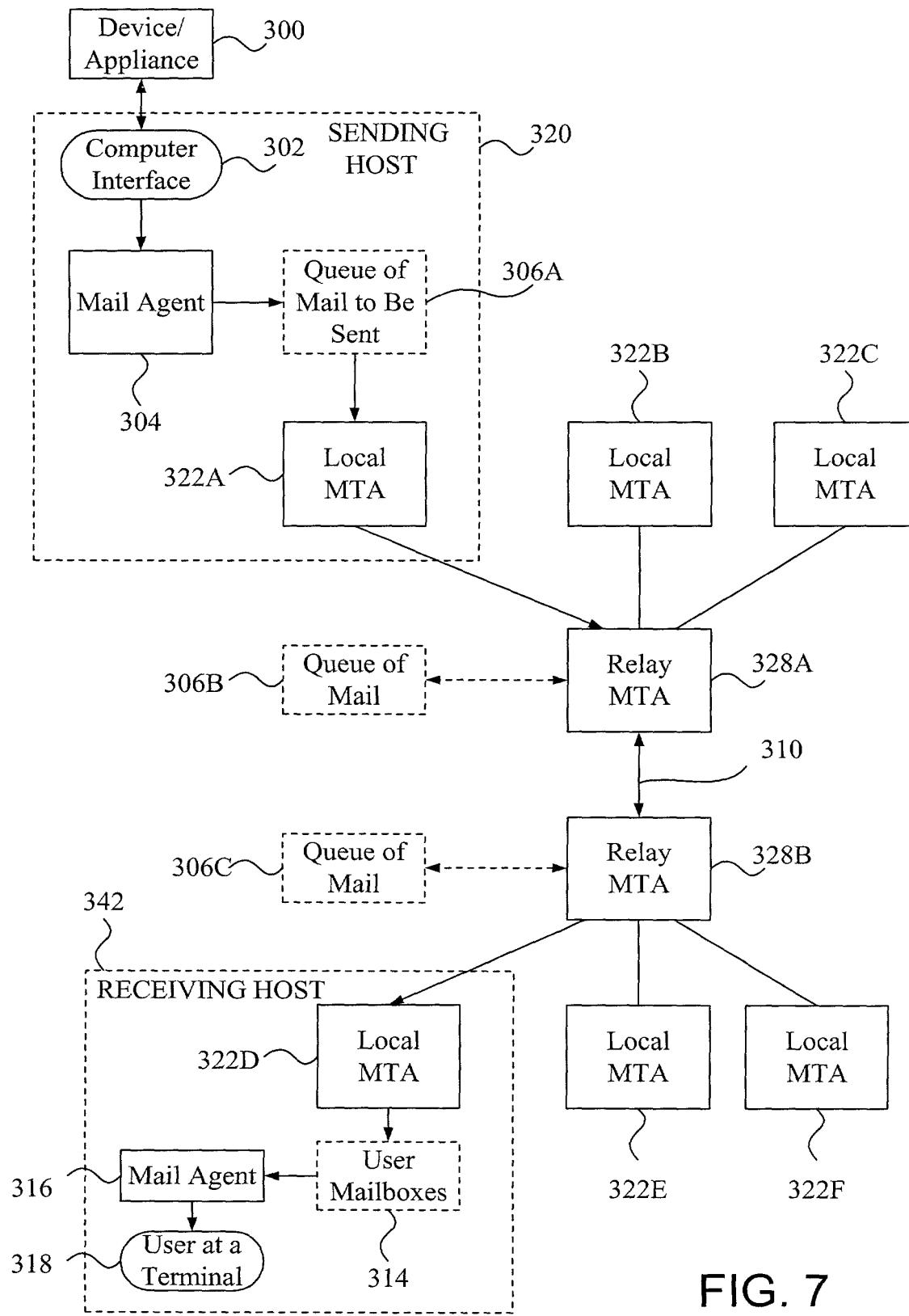


FIG. 7

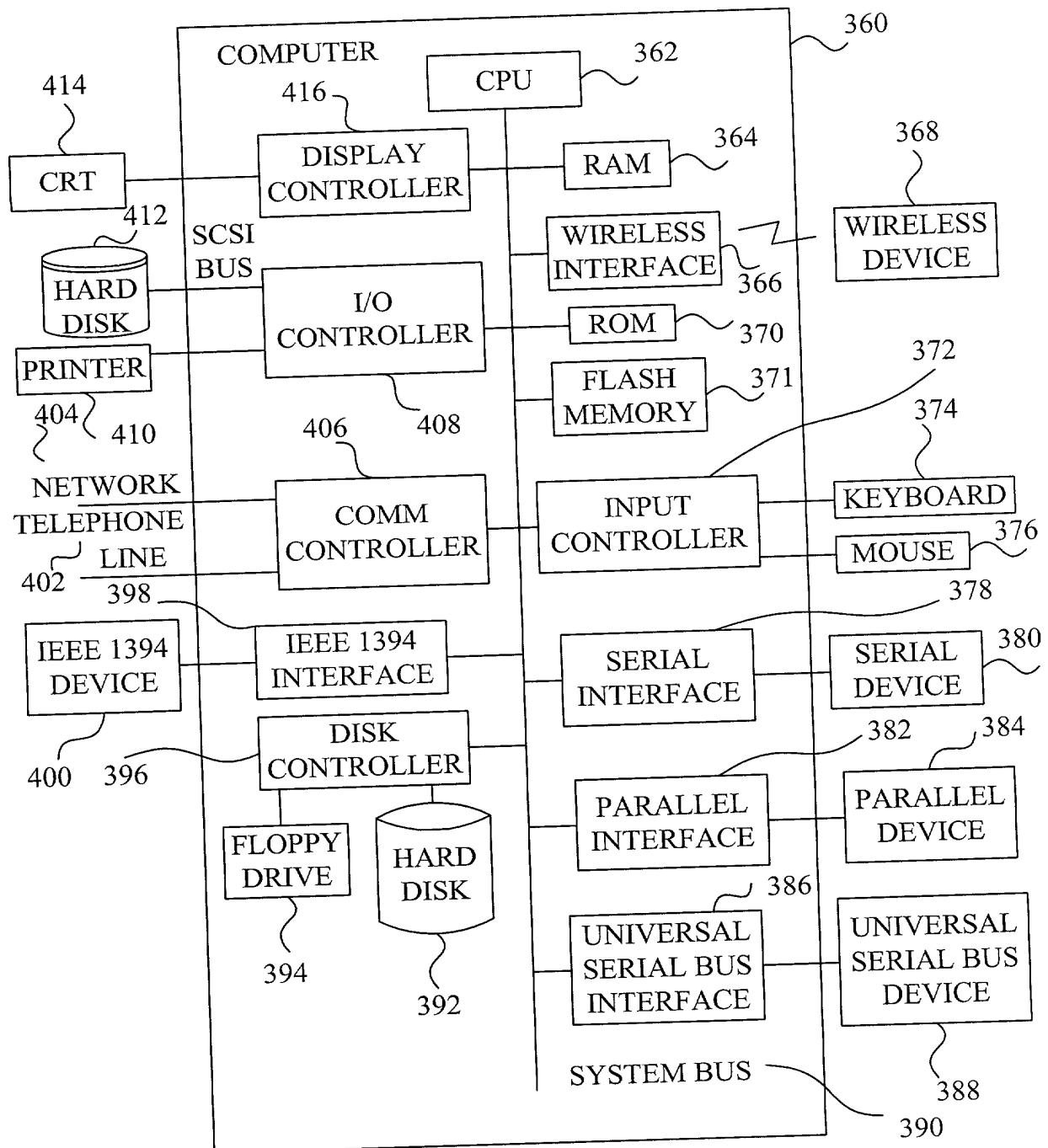


FIG. 8

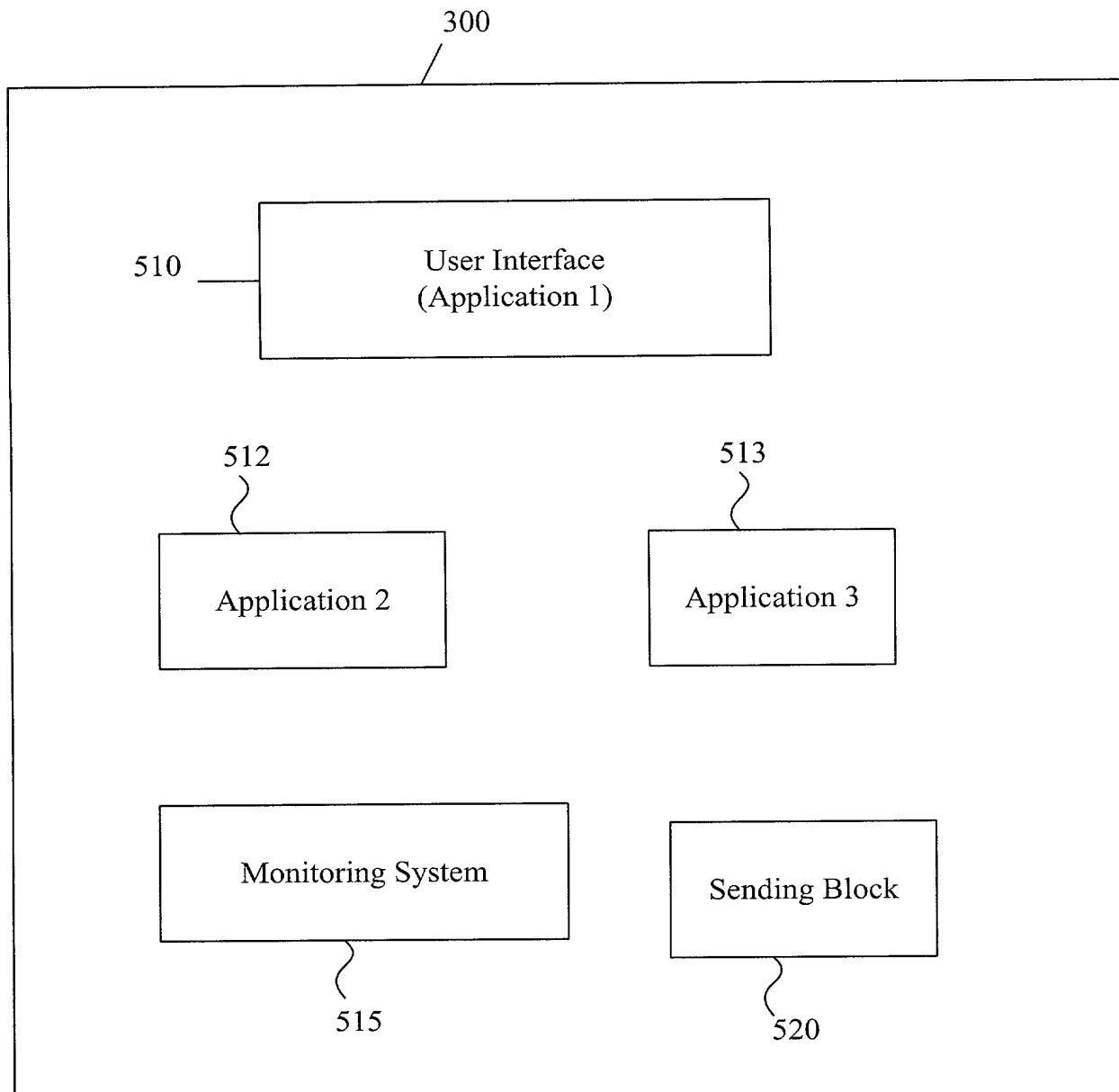


FIG. 9

FIG. 10

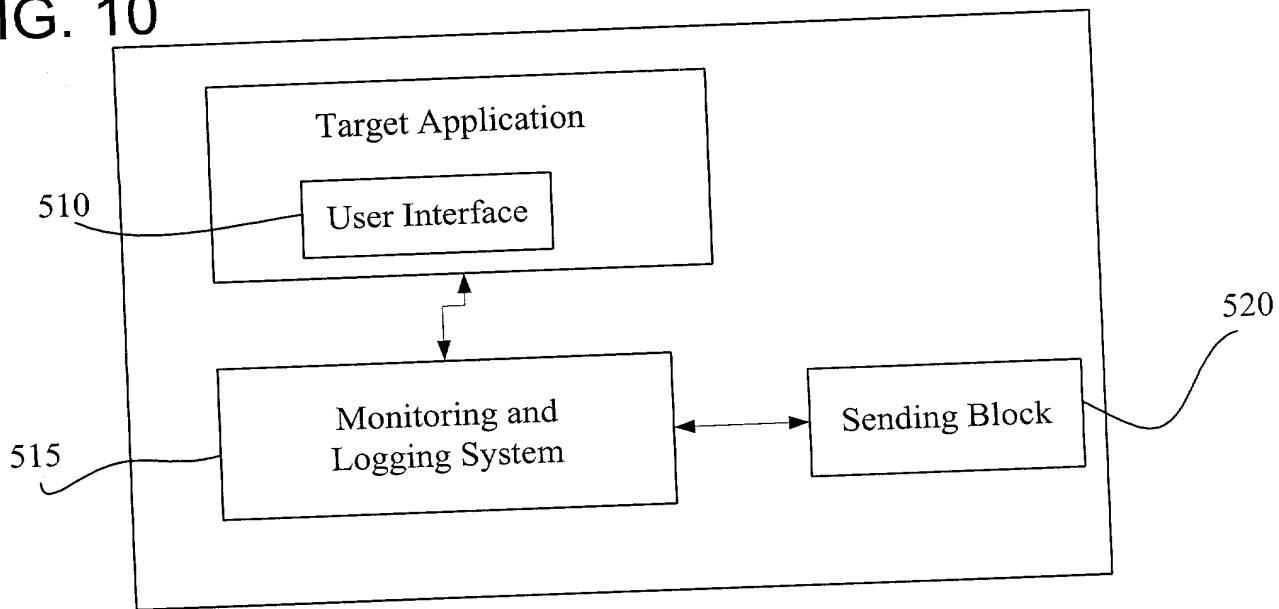
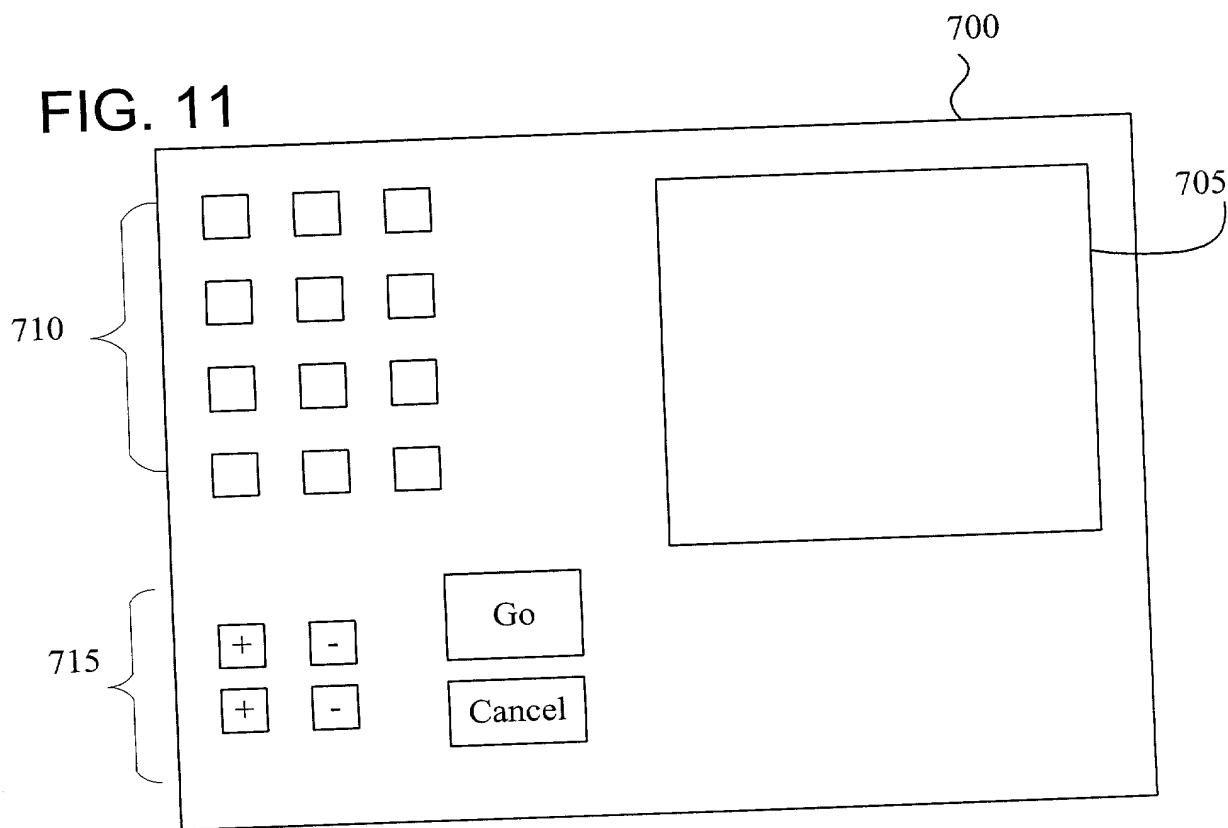


FIG. 11



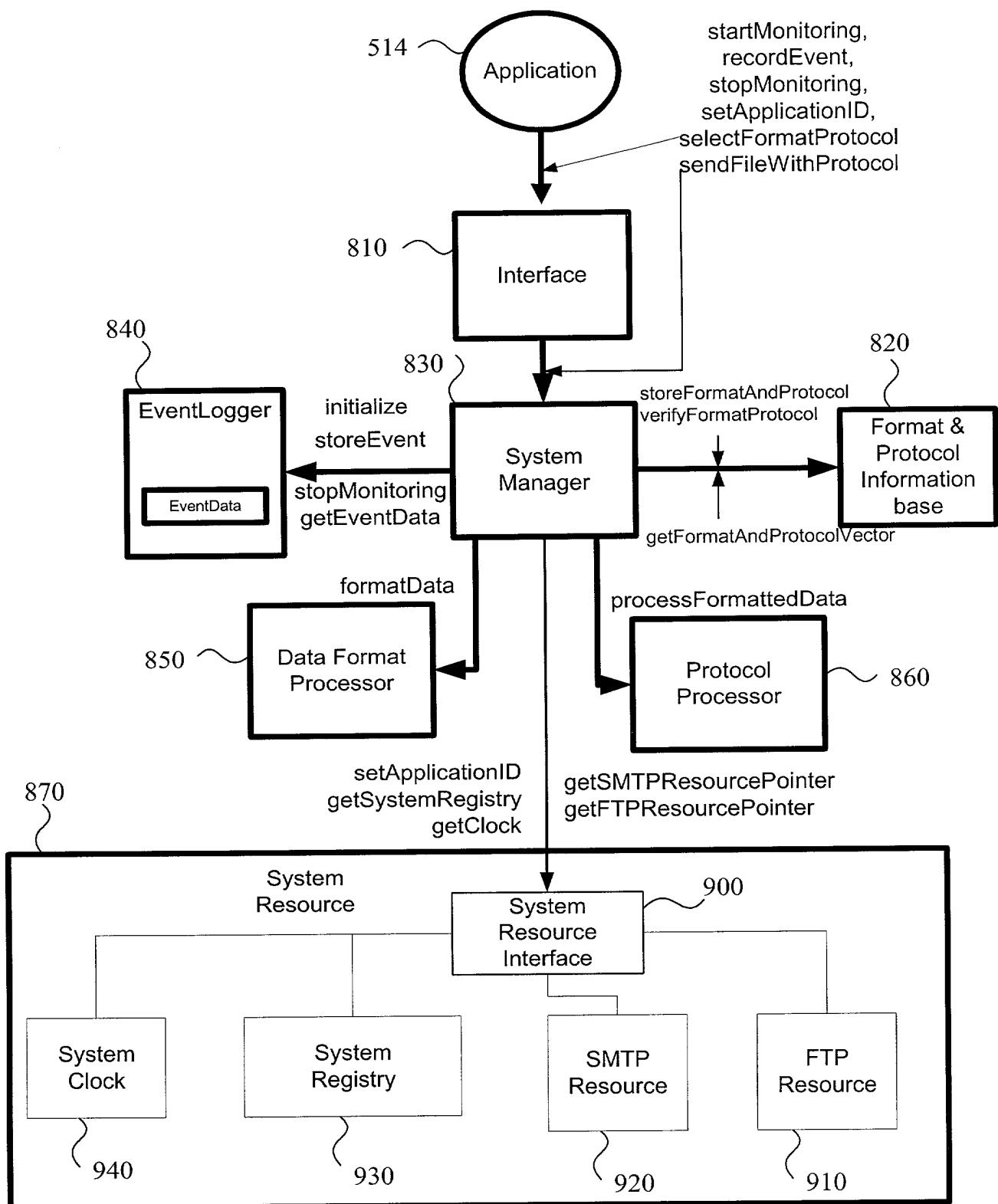


FIG. 12A

Return Value	Function Name	Description
bool	getNextSession	Returns false when there is no more session; true otherwise
string	getFileName	Returns file name for the EventData
map<string, string>	getSessionInformation	Returns the map. Keys are UserID, ApplicationID, CumulativeSessionNumber, StartTime, and Duration.
map<string, vector<string>>	getSessionEventData	Returns the map. Keys are EventName and EventTiming. The values of EventTiming vector are in the unit of 10th of a second converted from unsigned integer to string.

FIG. 12B

Return Value	Function Name	Description
bool	getNextLine	Returns one line of string data as an out parameter string. The function returns true if there is a line; false if no more line exists with empty string.
string	getFileNameWithSuffix	Returns file name for the data with suffix if applicable
enum	getDataType	Returns the data type, BINARY or TEXT

FIG. 12C

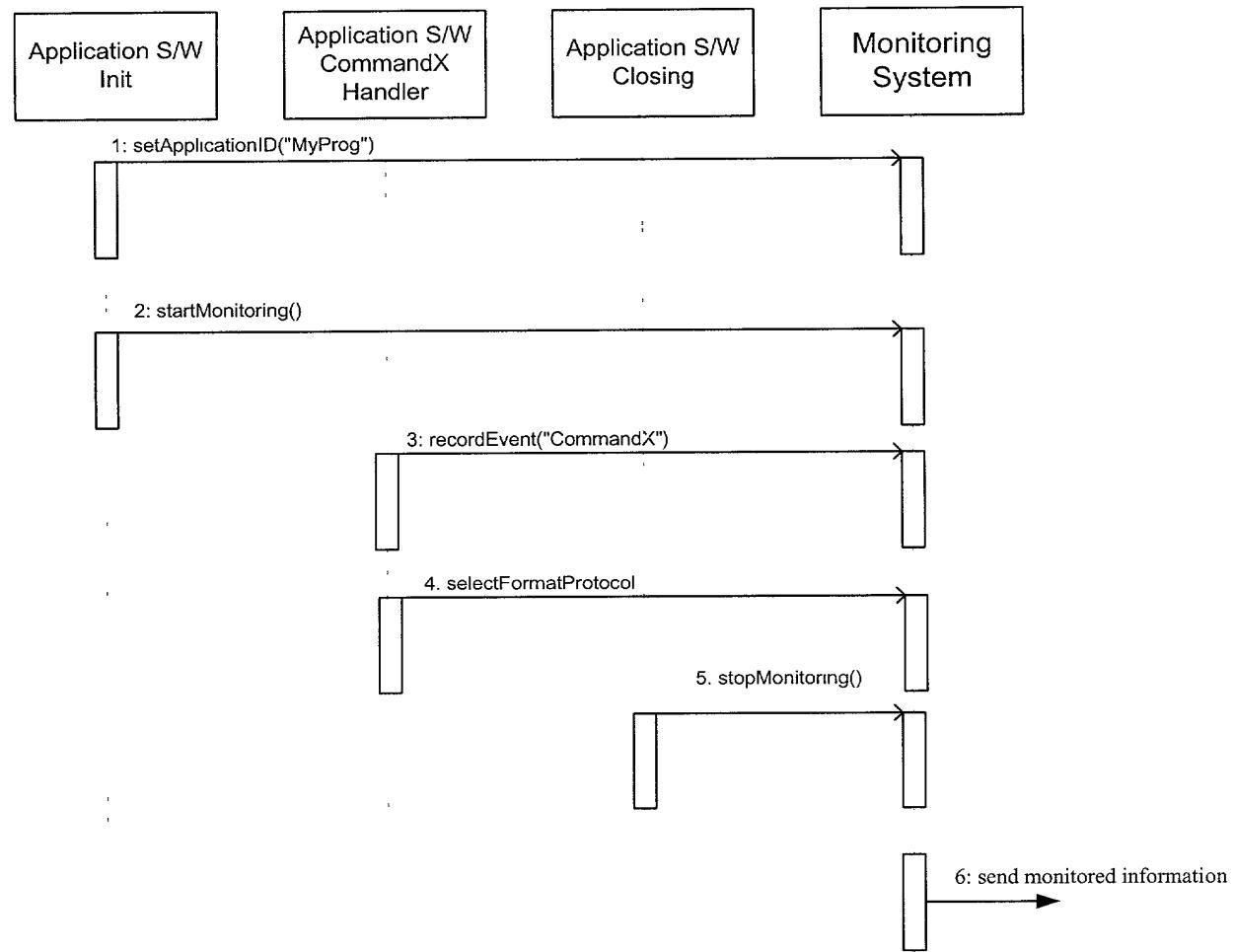


FIG. 13A

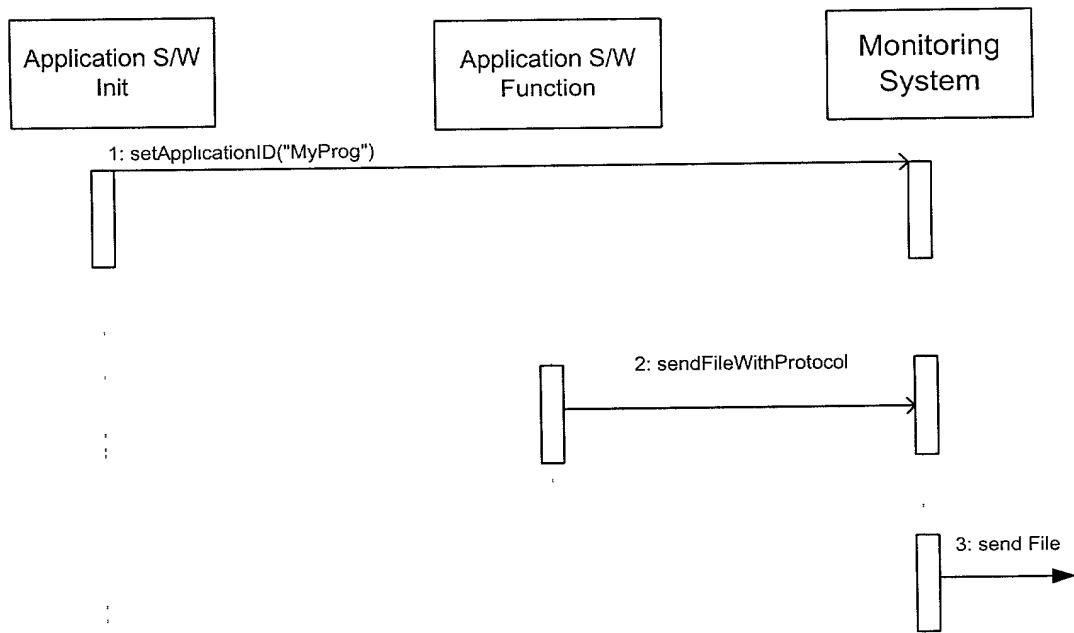


FIG. 13B

100 200 300 400 500 600 700 800 900 1000

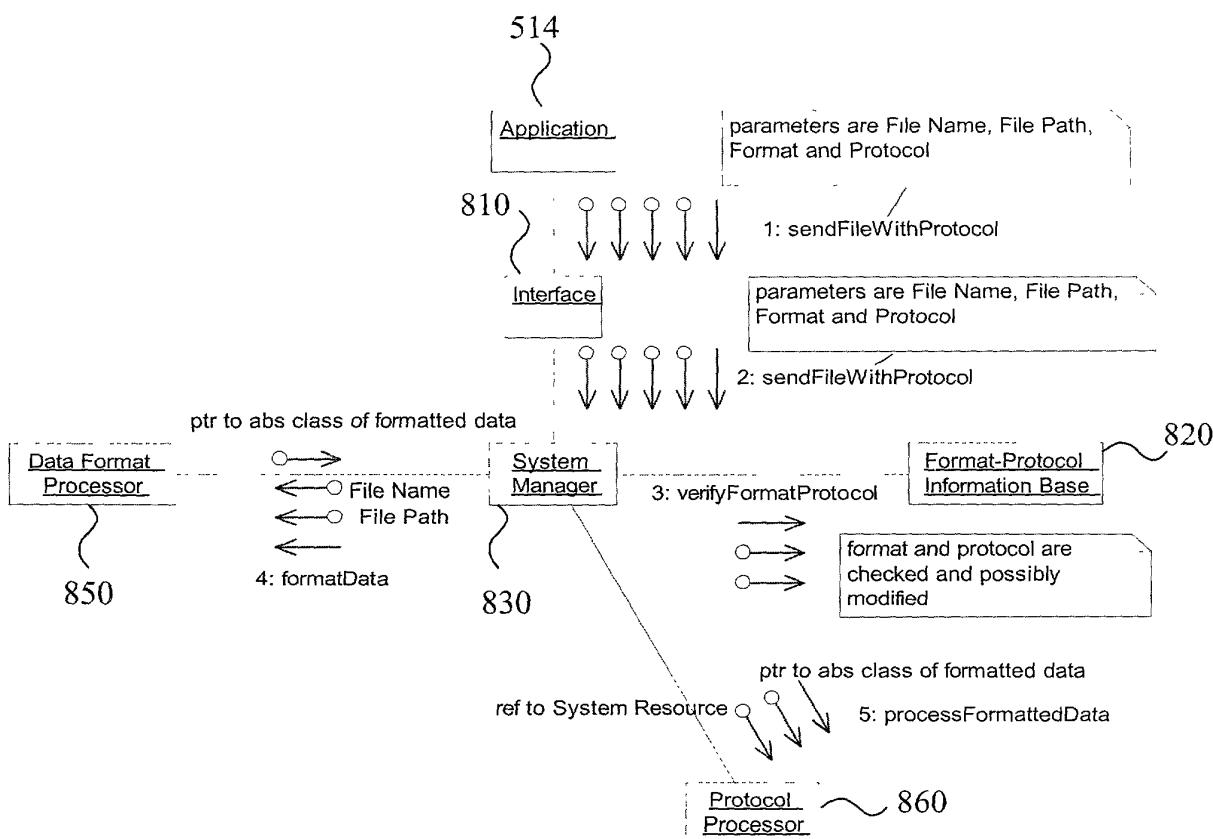


FIG. 14

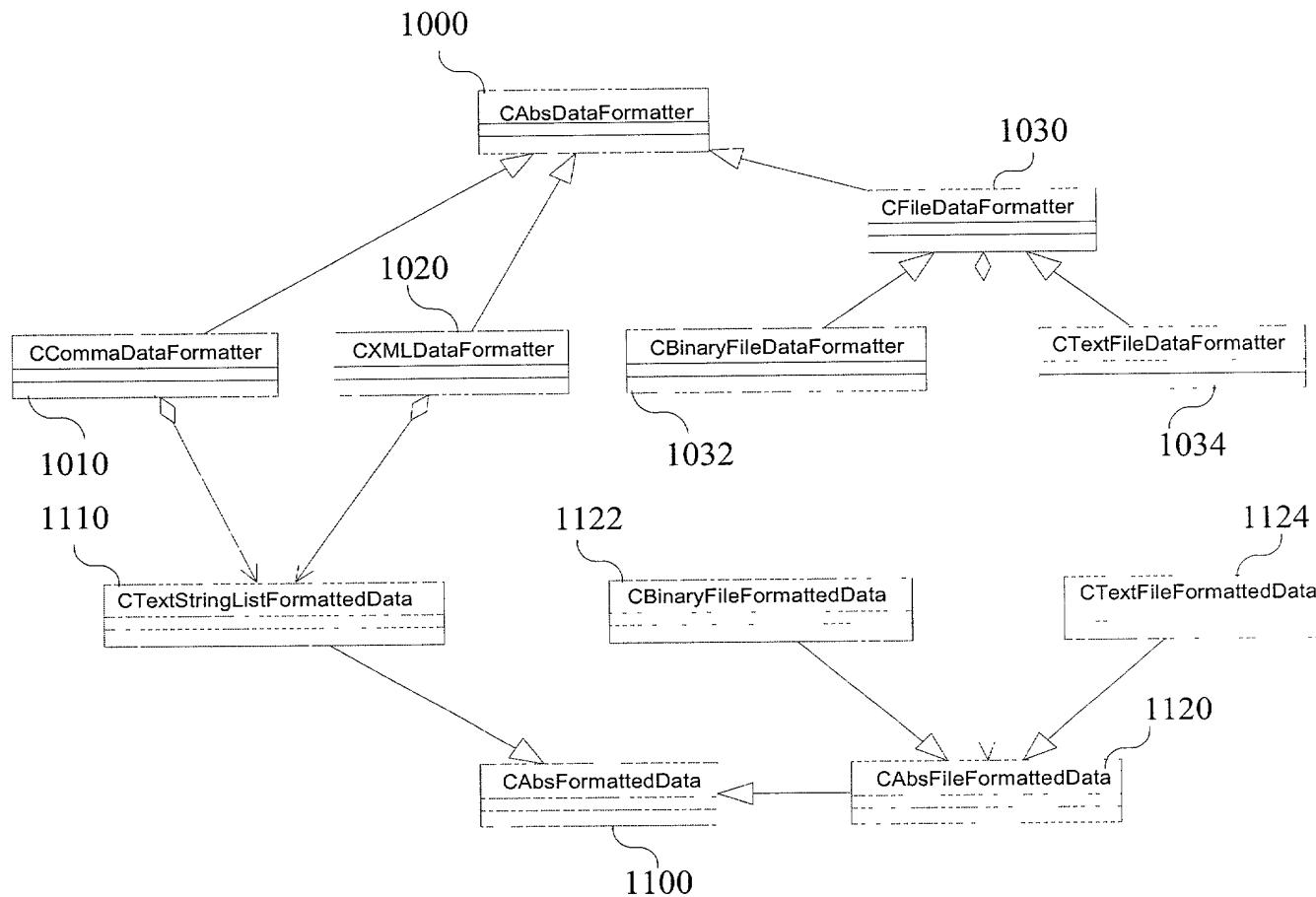


FIG. 15

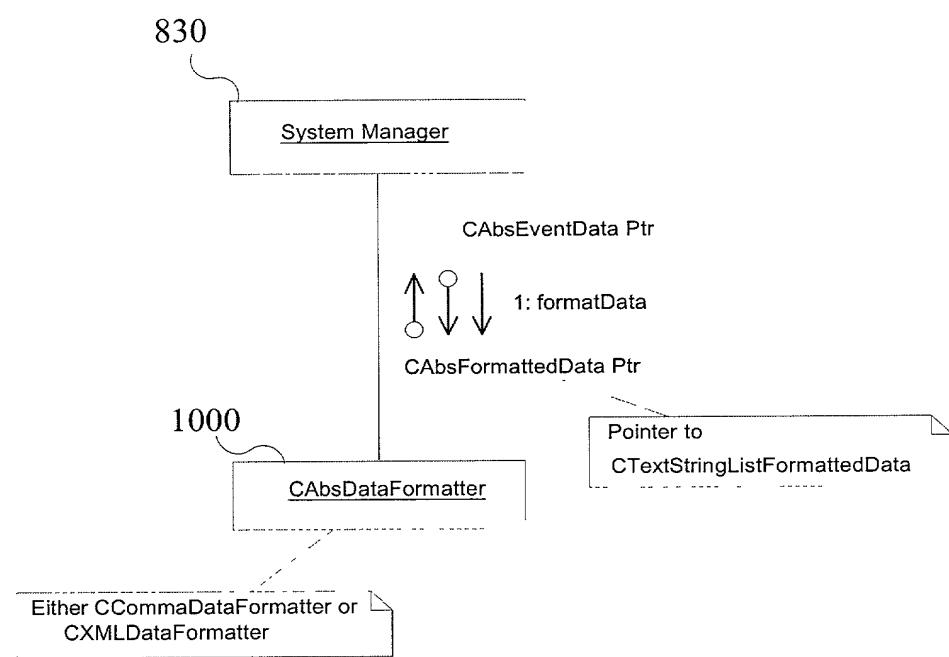


FIG. 16

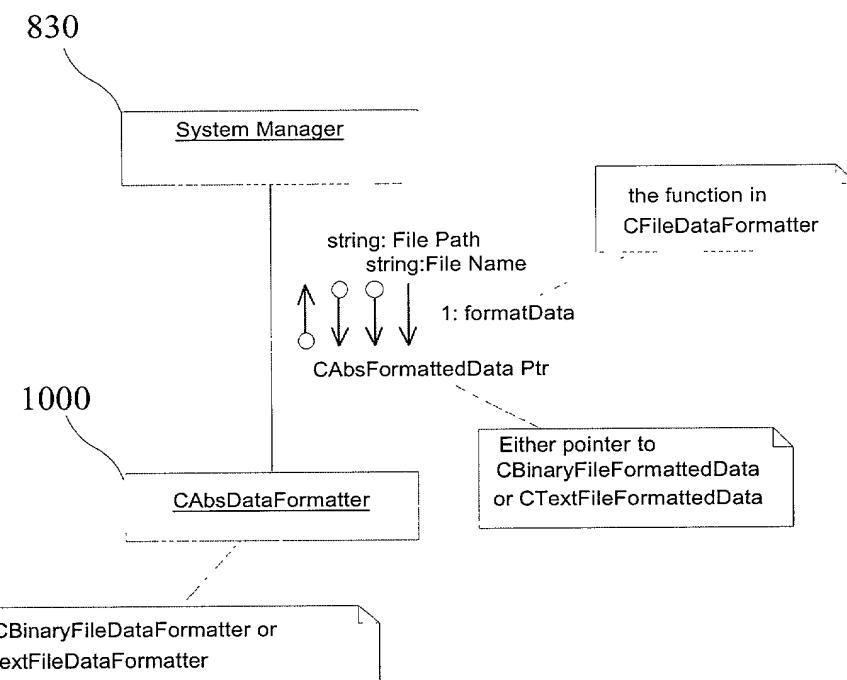
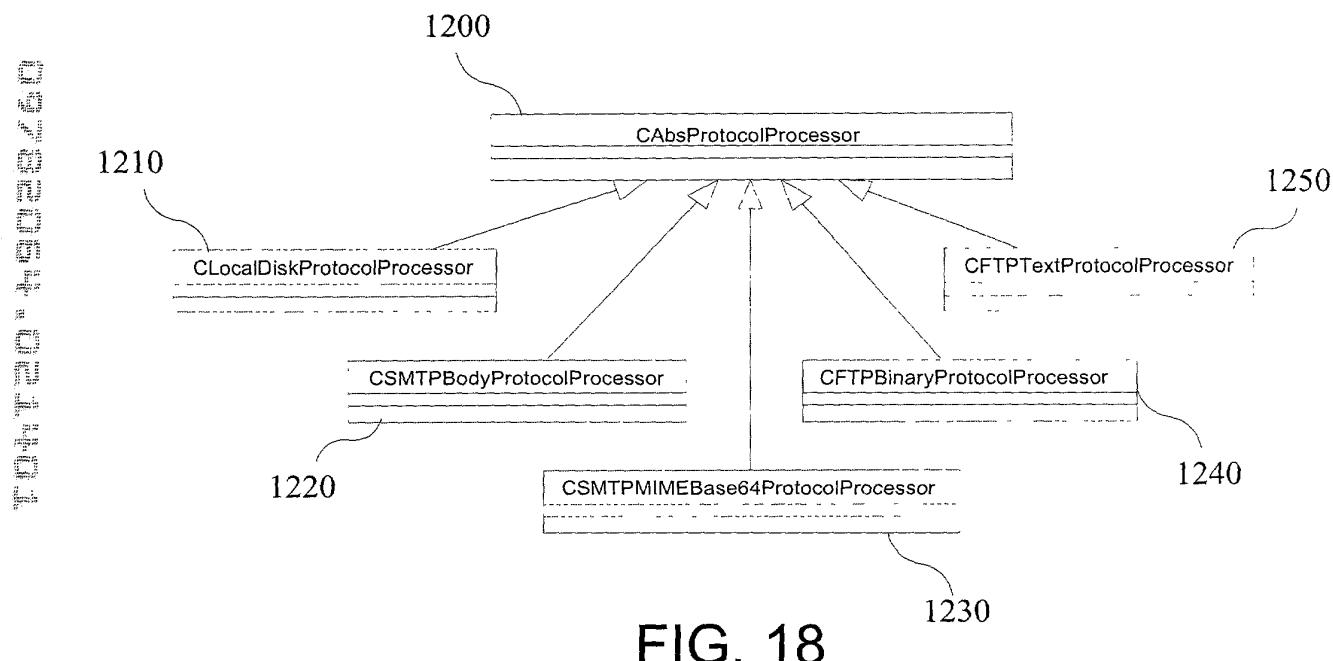


FIG. 17



**FIG. 18**

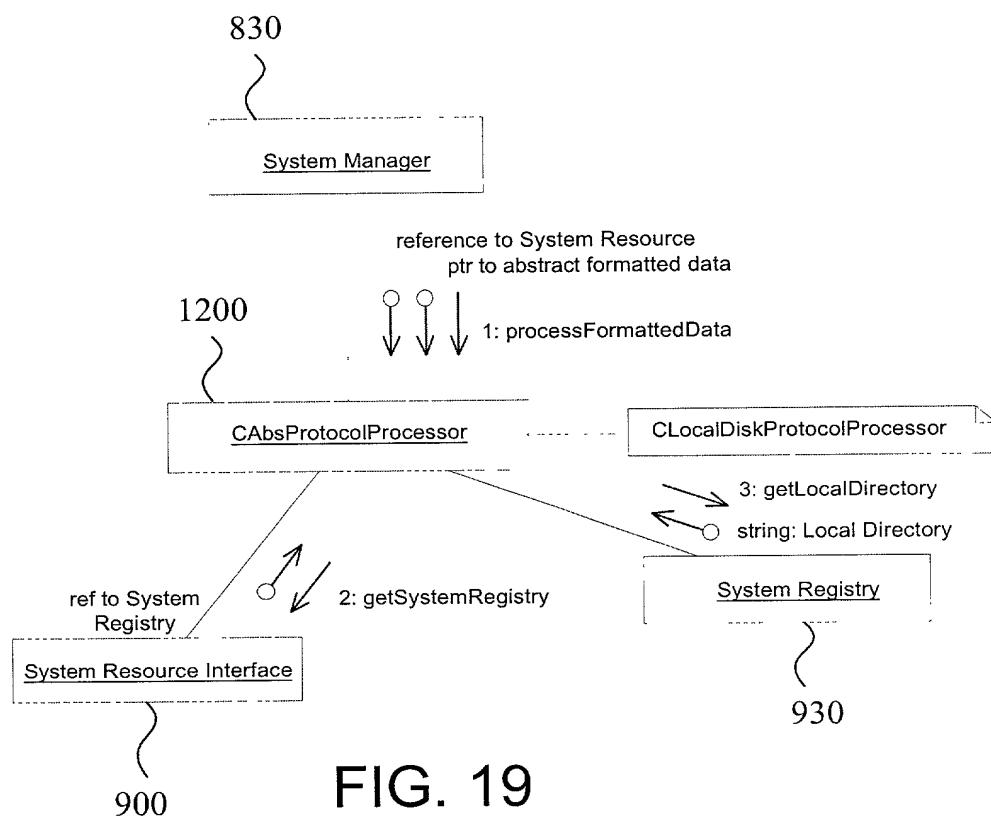


FIG. 19

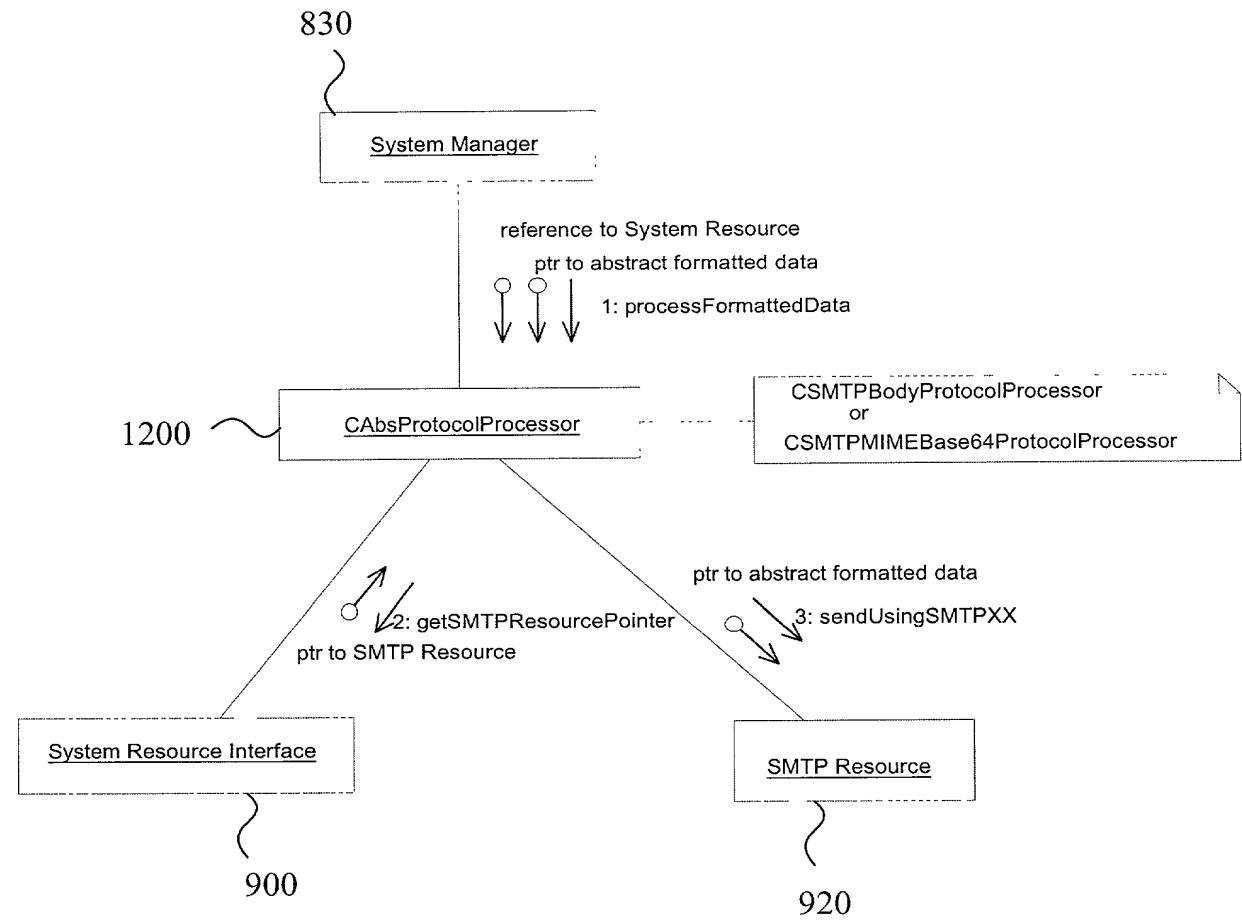


FIG. 20

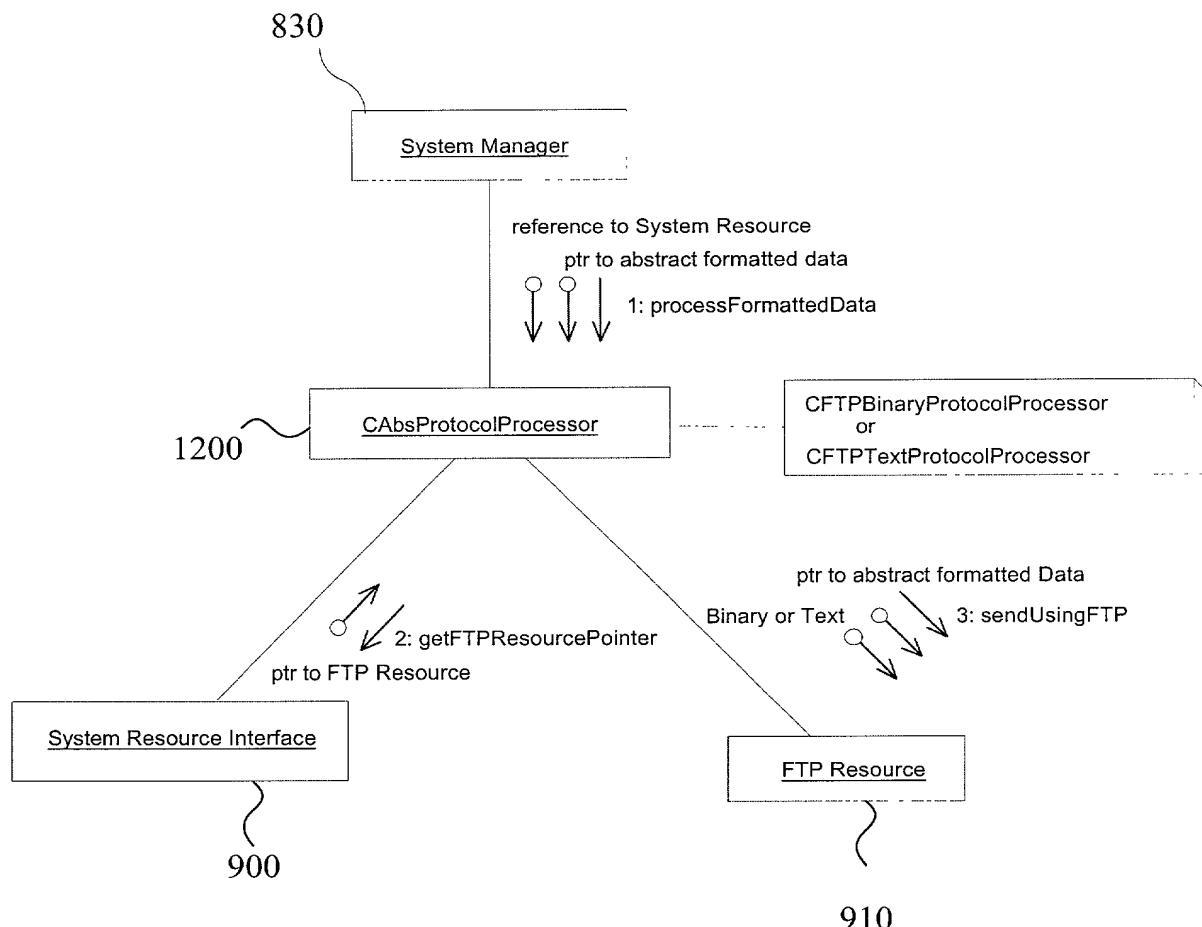


FIG. 21

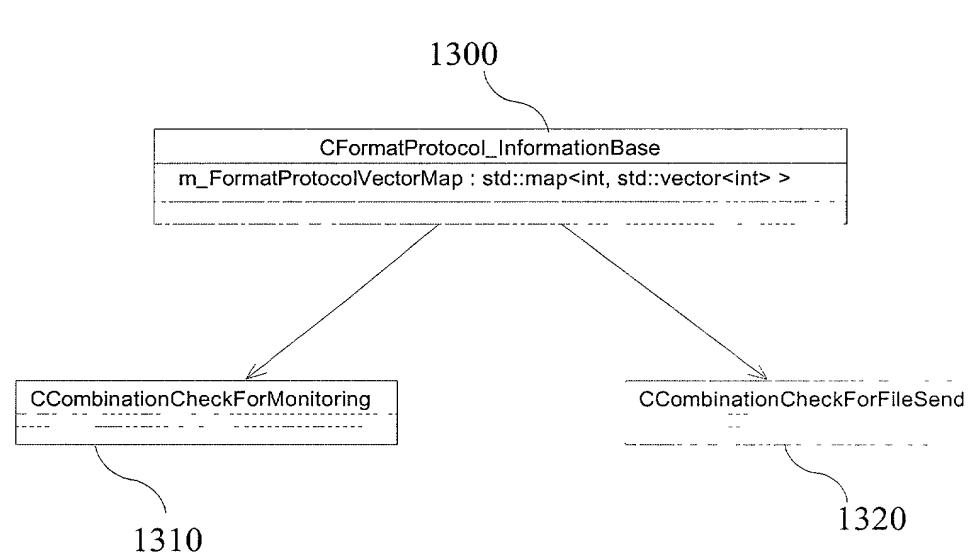


FIG. 22

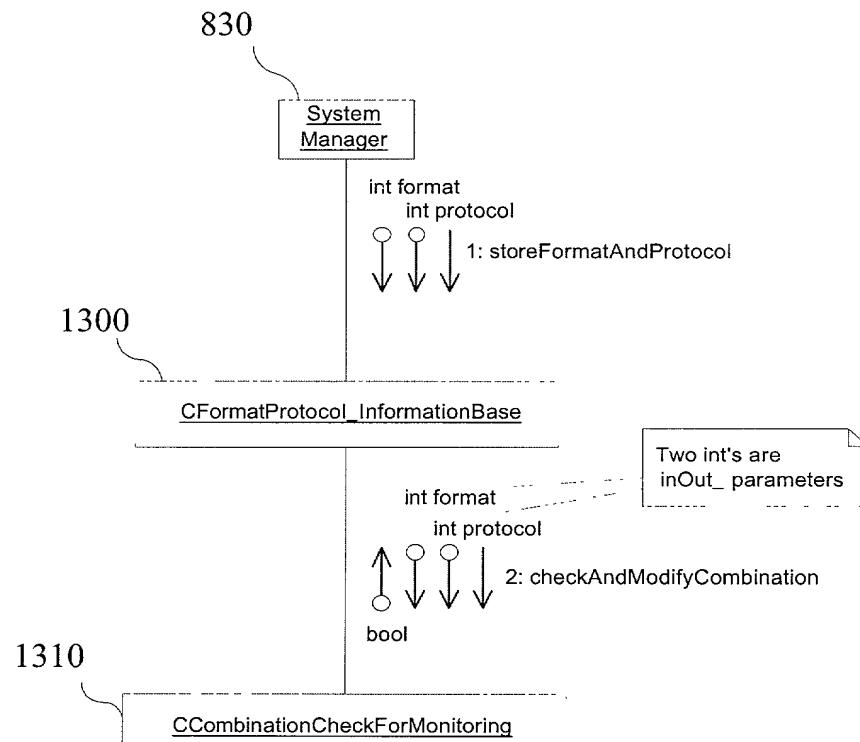


FIG. 23

1000 900 800 700 600 500 400 300 200 100 0

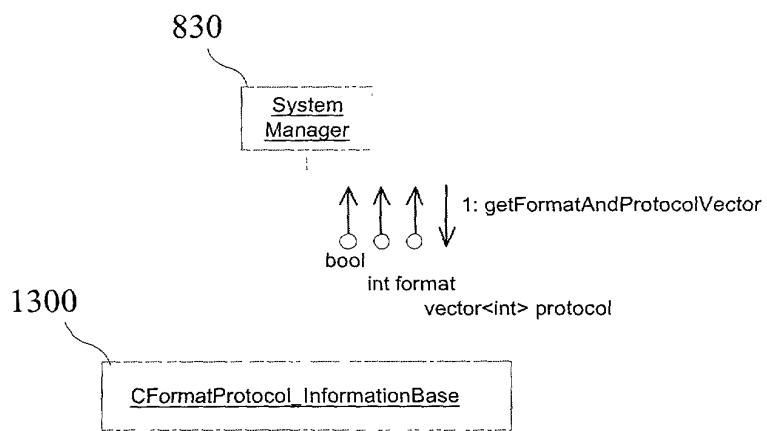


FIG. 24

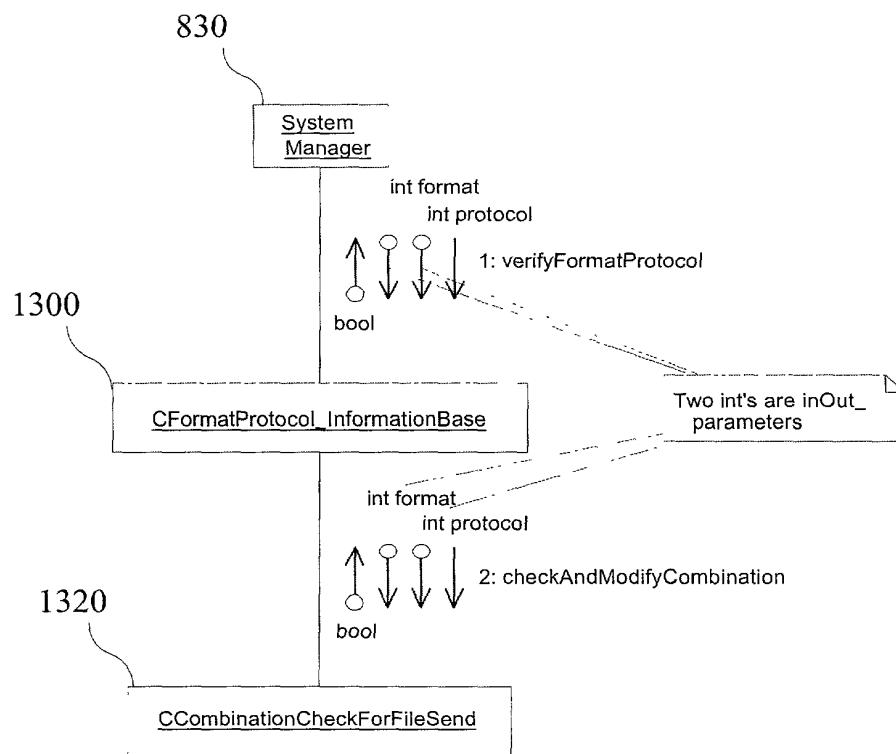


FIG. 25

Key	Value
10	{ 1 10 30 100 105 }
20	{ 1 10 30 100 105 }

FIG. 26A

1. Set return-bool to true
2. Use find function of the Map for inOut\_nFormat
3. If returned iterator is end (Not found), set inOut\_nFormat to the default value(10) and set return-bool to false
4. get the Set value for the key format
5. Use the find function of the Set for inOut\_nProtocol
6. if returned iterator is end (Not found), set inOut\_nProtocol to default (1) and set return-bool to false
7. return return-bool

FIG. 26B

Key	Value												
1	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>1</td><td>1</td></tr><tr><td>10</td><td>10</td></tr><tr><td>30</td><td>30</td></tr><tr><td>100</td><td>100</td></tr><tr><td>105</td><td>105</td></tr></tbody></table>	Key	Value	1	1	10	10	30	30	100	100	105	105
Key	Value												
1	1												
10	10												
30	30												
100	100												
105	105												
5	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>1</td><td>1</td></tr><tr><td>10</td><td>30</td></tr><tr><td>30</td><td>30</td></tr><tr><td>100</td><td>105</td></tr><tr><td>105</td><td>105</td></tr></tbody></table>	Key	Value	1	1	10	30	30	30	100	105	105	105
Key	Value												
1	1												
10	30												
30	30												
100	105												
105	105												

FIG. 27A

1. Set return-bool = true
2. Use find function of the Map for inOut\_nFormat
3. If returned iterator is end, set inOut\_nFormat to the default value (5) and set return-bool to false
4. Get the Map corresponding to the key format
5. Use the find function of the Map for inOut\_nProtocol
6. if returned iterator is end{  
    set inOut\_nProtocol to default (105)  
    and set return-bool to false  
}  
else {  
    return-bool = (inOut\_nProtocol EQ  
                  (Value field corresponding to  
                  inOut\_nProtocol))  
    logical-AND return-bool.  
    set inOut\_nProtocol =  
        (Value field corresponding to inOut\_nProtocol).  
}  
7. return return-bool

FIG. 27B